

# Смертельная схватка: .Net против J2EE

*Имеет ли одна из платформ, поддерживающих Web-службы, преимущество над другой?*

Server-Side Java

Джонатан Лури, Ясон Белангер

Java 2 Platform, Enterprise Edition (J2EE) и .Net являются конкурирующими технологиями, каждая из которых позволяет создавать Web-службы (Web services). В настоящей статье Jonathan Lurie и R. Jason Belanger описывают технологию Web-служб и приводят сравнение основных компонент платформ J2EE и .Net. Вооружившись этой информацией, Вы сможете понять, какую стратегическую помощь могли бы оказать Web-службы Вашей компании. (2600 слов)

Несмотря на то, что в настоящее время ведутся жаркие споры вокруг преимущества одной из платформ (J2EE и .Net), многие полагают, что это не более, чем маркетинговая война. Неважно, так это или нет, но очевидно, что результат этих споров существенно повлияет на эволюцию программного обеспечения в будущем. Руководители Sun Microsystems и Microsoft вложили значительные средства в раскрутку своих платформ и хотят получить соответствующую отдачу. Если Microsoft проиграет эту схватку, у нас появится возможность свободного выбора операционной системы на свой вкус. Победа Sun позволит программному обеспечению выполняться на любых операционных системах и приведет к тому, что доминирование Microsoft в области операционных систем понемногу исчезнет, в то время как на рынке появятся другие операционные системы, на которых сможет выполняться то же самое программное обеспечение. С другой стороны, если победит Microsoft, она еще более укрепит свои технологии в качестве фактических стандартов

на ближайшее обозримое будущее.

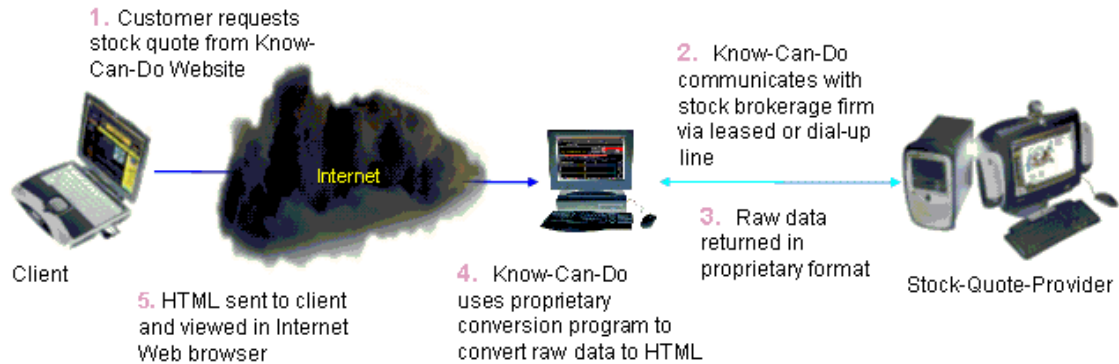
В данной статье мы сравниваем технологии J2EE и .Net, чтобы помочь вам решить, в какую из них стоит вкладывать деньги.

## 1. Что было до Web-служб

Многие интернет-аналитики, оглядываясь на короткий период, предшествовавший обвалу дот-ком доменов, отмечают, что многие из них просто дублировали друг друга. Наибольшее дублирование имело место в области структуры Web-сайта. Дело в том, что эти, теперь уже древние Web-сайты, пытались снабдить посетителя огромным количеством информации; информации, относящейся не к основной области деятельности компании, а выставленной, скорее, из-за желания компании выглядеть более привлекательно. Компании, предоставляющие такие разнообразные услуги, включающие информацию о погоде, курсы акций, новости, почтовые услуги, и т.д., обычно не сами обеспечивают их. Следовательно, им приходится покупать права на использование первичных данных, а также на то, чтобы представить эти данные удобным для просмотра способом. После получения прав на использование первичных данных, компаниям приходилось создавать дорогие и требующие больших затрат времени программы, которые преобразовывали первичные данные в формат, пригодный для показа пользователю (обычно — HTML).

Например, предположим, что компания “Know-Can-Do” на своем Web-сайте предлагала информацию о курсе акций. Для этого ей необходимо было получить первичные данные от их поставщика, скажем, компании “Stock-Quote-Provider”. Компания “Know-Can-Do” в типичном случае получала данные от “Stock-Quote-Provider” с помощью некоей специально разработанной технологии (например, специально установленного программного обеспечения), а также дорогостоящих аппаратных средств (например, выделенной линии). Более того, “Know-Can-Do” приходилось создавать приложения для преобразования первичных данных в HTML. Этот процесс проиллюстрирован на рисунке 1. Обратите внимание на светло-голубую линию, которая обозначает взаимодействие между “Know-Can-Do” и “Stock-Quote-Provider”; используемая для этого взаимодействия технология весьма дорога, так как специально создана для данной системы.

## Смертельная схватка: .Net против J2EE



**Рисунок 1. Движение данных и управление ими с помощью специально разработанной технологии.**

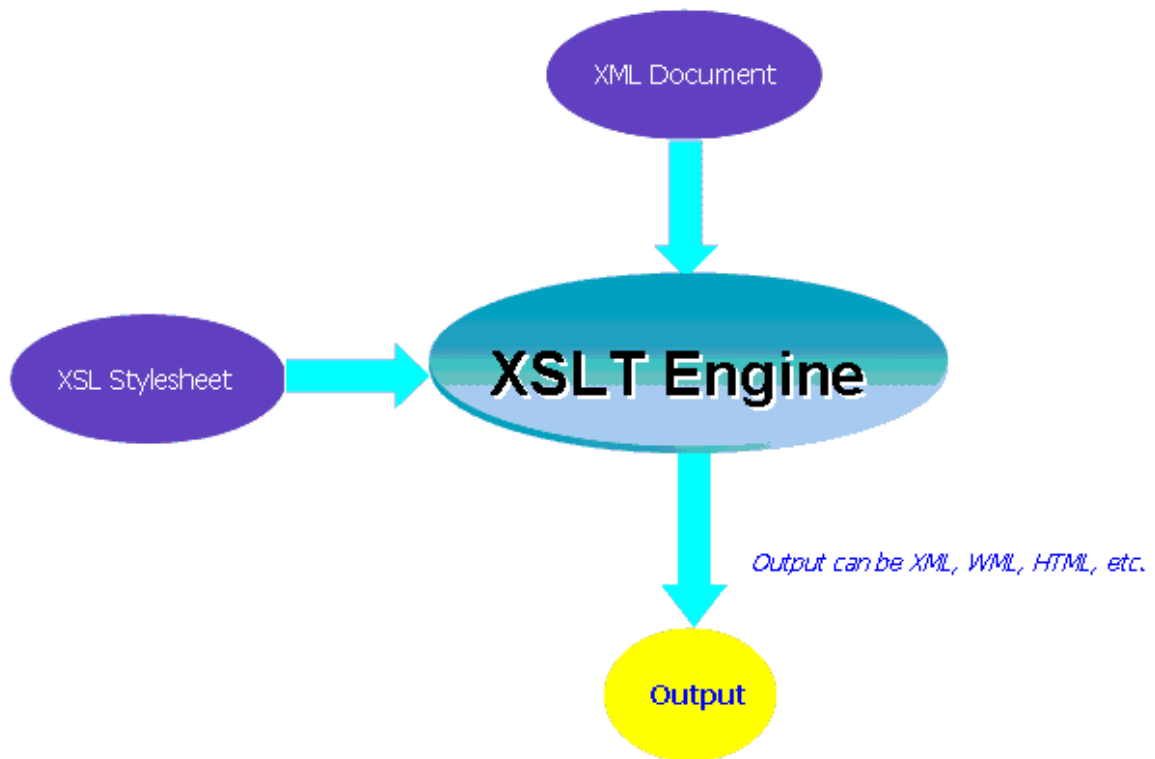
Поскольку компания “Know-Can-Do” хотела бы конкурировать с такими доминирующими Web-порталами как AOL, MSN (Microsoft Network), Yahoo! и Excite, необходимость предоставлять посетителям больше информации стремительно увеличивалась. Чтобы оставаться конкурентоспособной, “Know-Can-Do” была вынуждена добавить услуги по предоставлению информации о погоде и новостей. Оказалось, что использовать для этого модель, представленную на рисунке 1 весьма неэффективно, так как получение данных от различных поставщиков происходит различными, несовместимыми способами и этот процесс становится неуправляемым как с точки зрения технологии, так и с точки зрения стоимости. Необходимо было добиться меньших затрат для получения первичных данных, и кроме того, “Know-Can-Do” нуждалась в более простой технологии для преобразования данных к стандартному виду (например, HTML, WML (Wireless Markup Language), Voice XML).

И вот теперь давайте рассмотрим, что нам предлагает стратегия Web-служб.

## 2. Что такое Web-служба?

Web-службы появились как решение, позволяющее стандартным способом получать необходимые данные, без какого-либо специально для этого созданного программного или аппаратного обеспечения. Краеугольным камнем технологии Web-служб является их способность передавать данные от поставщика к потребителю, используя всего лишь повсеместно распространенный HTTP-протокол; при этом в качестве формата

данных используется XML. Использование XML в качестве формата данных существенно облегчает преобразование первичных данных в формат, пригодный для просмотра пользователем. Такое простое преобразование, не требующее сложных программ для разбора данных, обеспечивается языком XSLT (Extensible Stylesheet Language Transformations). Вышесказанное иллюстрируется рисунком 2.



**Рисунок 2. Высокоуровневая схема XSL-преобразования.**

Официальный документ фирмы Sun определяет Web-службу следующим образом:

Web-служба – это приложение, которое получает запросы от других систем через интернет или интранет, используя для этого коммуникационные технологии, независимые от платформы и поставщика.

В документе "Defining the Basic Elements of .Net" Microsoft определяет Web-службу так:

Web-службы, основанные на XML, служат для обмена данными между

приложениями, и что более важно, позволяют вызывать другие приложения независимо от того, как эти приложения устроены, на какой платформе они работают и какие устройства используются для доступа к ним.

Из этих определений следует один приятный вывод: Sun и Microsoft неявно соглашаются друг с другом по поводу определения Web-службы. На чисто интуитивном уровне Web-служба – это некий сервис, доступ к которому осуществляется через интернет. Более детально можно сказать, что Web-служба вызывается с помощью протокола HTTP и возвращает данные в формате XML.

### **3. Концептуальный поворот**

Появление Web-служб влечет серьезные изменения в парадигме разработки программного обеспечения. Пока некоторые компании все еще оспаривают право Web-служб на существование, другие компании, такие как “Concord EFS”, зарабатывают миллионы долларов, с помощью Web-службы, обрабатывающей кредитные карточки. Web-службы подталкивают нас к разбиению больших приложений на небольшие независимые части, которые могли бы существовать в качестве Web-служб. Такая модель, возможно, заменит существующую парадигму в соответствии с которой разбиение происходит на динамические библиотеки (DLL, Dynamic Link Library) и COM (Component Object Model) объекты.

На самом деле, Web-службы и библиотеки DLL весьма похожи. И те, и другие аккумулируют некий набор взаимосвязанных функций; например, бизнес-логику или логику доступа к базе данных. Тем не менее, между ними существует и существенная разница. Во-первых, Web-службы доступны через протокол HTTP, что позволяет любому Web-клиенту вызвать их. В случае DLL все обычно происходит по-другому, и клиент находится в том же интранете, что и DLL. Таким образом, Web-службы открывают новую эру распределенных вычислений. Во-вторых, Web-службы возвращают данные клиенту в формате XML. DLL обычно возвращают типы данных, специфические для используемого языка программирования.

Эти отличия между Web-службами и их предшественниками (DLL) определяются следующими тенденциями в программной индустрии, проявившимися до появления Web-служб:

1. Принятие HTTP как стандартного протокола, с помощью которого осуществляется доступ в интернет;
2. Принятие XML де-факто как стандарта для передачи данных.

Эти две тенденции обеспечили базис, на котором были построены Web-службы. Рисунок 3 иллюстрирует как компания “Know-Can-Do” могла бы использовать Web-службы. Заметьте, что теперь ей больше не требуются ни выделенные линии, ни специально созданный формат обмена данными.

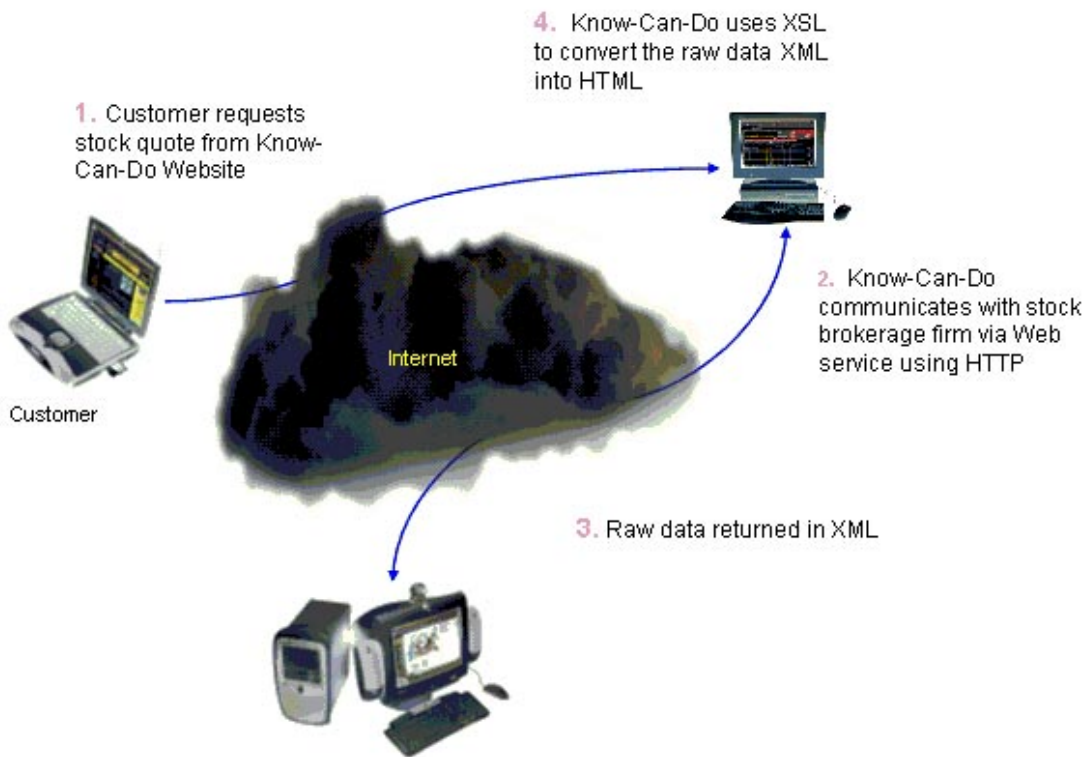


Рисунок 3. Превью пример, спроектированный с помощью технологии Web-служб.

## 4. Java 2 Platform, Enterprise Edition

Часто бывает, что даже те, кто понимают технологию Web-служб, не понимают сущность платформы J2EE. Многие люди (исключая разработчиков) полагают, что

## *Смертельная схватка: .Net против J2EE*

J2EE – это некий продукт, который вы покупаете у Sun так же, как вы покупаете .Net у Microsoft. На самом деле, напротив, J2EE – это всего лишь набор спецификаций, каждая из которых устанавливает, как должны работать различные функции из состава J2EE. Например, спецификация Java Transaction Service (JTS) определяет, как создать сервис, поддерживающий распределенные транзакции. Sun предлагает свои образцы реализаций этих спецификаций, которые можно использовать для проверки совместимости со спецификациями.

Возникает вопрос: если вы не покупаете J2EE у Sun, то как же тогда Sun зарабатывает деньги? Sun зарабатывает их, выдавая лицензии независимым поставщикам программного обеспечения (independent software vendors, ISV), которые реализуют данные спецификации и продают в виде готовых продуктов на рынке. Таким образом, если вам надо купить реализацию J2EE вы можете это сделать у одного из поставщиков программного обеспечения, который разработал J2EE-совместимую реализацию. Для того, чтобы найти список таких реализаций, можно сходить на <http://java.sun.com/j2ee/compatibility.html>

Для того чтобы ознакомиться со списком компаний, получивших лицензию, можно сходить на <http://java.sun.com/j2ee/licensees.html>. Вы увидите, что Microsoft в этом списке отсутствует - это следствие известной тяжбы, которая закончилась выплатой 20 миллионов долларов.

Хотя технология сервлетов в J2EE и не проектировалась с учетом будущего использования Web-служб, она поддерживает эту технологию. Сервлет выполняет все обработку вызова, включая обращение к Enterprise JavaBeans (EJBs), которые возвращают результирующие данные сервлету. Далее сервлет формирует ответ (response), который он заворачивает в XML для доставки клиенту. Новый продукт Sun — Java Web Services Developer Pack (WSDP) содержит все необходимое для создания Java Web-служб. Он содержит в себе Java XML Pack, который позволяет разработчику абстрагироваться от низкоуровневого разбора XML, а также JavaServer Pages (JSP) Standard Tag Library 1.0, Ant 1.4.1, Java WSDP Registry Server 1.0 и Tomcat Java Servlet and JSP container 4.1.

## **5. Microsoft's .Net**

Прежде чем исследовать платформу Microsoft's .Net, мы должны понять, какие события стали поводом для ее появления. К концу 1995 года Microsoft стала перемещать свое основное внимание на интернет; дело в том, что компания слишком поздно вступила в игру в этой области, увлекшись Windows 95. В это время Netscape быстро завоевывала этот сегмент рынка, пока Microsoft силилась раскрутить собственный Web-браузер. Microsoft задействовала все свои колоссальные маркетинговые возможности (не будем вдаваться в споры о проявлении монополизма) и разрушила лидерство Netscape (которая тогда имела более 16 миллионов пользователей).

В это же время Microsoft вступила в схватку по раскрутке таких своих технологий, как Active Server Pages (ASP). Эта технология была достаточно эффективной, хотя и не достаточно зрелой; в конце концов, данная технология предоставляет среду для написания скриптов, что является шагом назад с точки зрения объектно-ориентированного подхода. Можно представить себе, с какой скоростью работали тогда команды разработчиков Microsoft, чтобы как можно скорее распространить свое программное обеспечение, напоминая о временах, когда Силиконовая Долина только родилась. В то время компания выпустила много таких технологий, и многие ранние технологии бесславно исчезли. Можно, например, вспомнить Active Documents – технологию для Visual Basic, которая позволяла разработчикам создавать Web-приложения без всякого дополнительного программирования. Эта технология умерла тихой смертью. Мы обязательно должны помнить о такого рода технологиях в процессе оценки .Net, чтобы понять, не является ли и .Net подобной фикцией.

Основная маркетинговая стратегия Microsoft – продвигать свои технологии всеми возможными средствами как можно дальше вширь и вглубь. Microsoft выходит на рынок с технологиями, которые лишь обеспечивают основной функционал. Некоторые из технологий проваливаются, а некоторые остаются и улучшаются до тех пор, пока не становятся стандартами отрасли. Поэтому главный вопрос здесь такой: .Net – это реальная технология, или Microsoft всего лишь блефует?

Справедливости ради, необходимо поблагодарить Microsoft за работу по доведению важности Web-служб до разработчиков. Компания провела такую огромную работу и затратила такие большие деньги на маркетинговые цели, что многие серьезно

## *Смертельная схватка: .Net против J2EE*

полагают, что именно Microsoft разработала технологии, позволяющие использовать Web-службы, хотя на самом деле, конечно, эта роль принадлежит Sun и ее технологии сервлетов. Конечно, дальше можно спорить о том, не была ли технология сервлетов ответом на технологию Microsoft ASP, и кто был первый, а кто – второй и кто был еще раньше, чем первый, и так до бесконечности. (Larry Seltzer возвращается к этому вопросу в публикации "Shadow Initiatives: .Net and Java" (ZDNet, January 2002)).

Теперь, когда разработчики осознают важность Web-служб, можно ожидать, что большее число компаний выделят Web-службы в своих приложениях и смогут продавать их другим компаниям. И здесь Microsoft захватила лидерство, одной из первых предоставив такие Web-службы с потенциально большой пользовательской базой. Например, разработчики могут использовать Web-службы NailStorm – строительные блоки для технологии .Net, выполняющие такие задачи, как обмен сообщениями, временная синхронизация и нотификация. Пожалуй, один из самых известных среди сервисов NailStorm — Microsoft's Passport, который обеспечивает идентификацию и аутентификацию пользователей. Согласно статистике Microsoft, в настоящее время он выполняет более 1.5 миллиарда операций аутентификации в месяц. Наиболее активно используемой Web-службой на сегодняшний день является Hotmail, пользователи, которого могут наблюдать логотип Microsoft's Passport на начальной странице.

Чтобы извлечь выгоду из процесса повсеместного внедрения Web-служб, Microsoft выступила с инициативой использования платформы .Net. С точки зрения архитектуры, .Net отличается от породившей ее технологии DNA (Distributed Internet Architecture), предлагая новые решения для технологии ASP, рассчитанные на использование Web-служб и реализованные в технологии ASP. Net. Эта технология предлагает среду с поддержкой полнофункционального языка программирования, а не только скриптов, как это было прежде. Возможно, наиболее существенным моментом внедрения .Net является то, что Visual Basic наконец-то стал объектно-ориентированным. Еще одна замечательная черта, которую не предлагает Sun, это способность ASP.Net создавать страницы в различных HTML-форматах, что позволяет разработчику не заботиться об поддержке различных версий HTML. Та же задача может быть решена и при помощи сервлетов, но для этого потребуется дополнительно кодировать вручную.

Новизна .Net в том, что .Net-приложения не компилируются в зависимый от платформы (native) код, например, специфический для архитектуры Intel. Вместо этого компиляция сейчас представляет собой процесс из двух шагов. Код, написанный разработчиком, компилируется в Промежуточный Язык Microsoft (Microsoft Intermediate Language (MSIL)). Затем с помощью среды Common Language Runtime (CLR) он компилируется в зависимый от платформы исполняемый код. Не правда ли, это что-то очень знакомое? Похоже, Microsoft умеет извлекать уроки и замечать, что происходит вокруг. .Net включает в себя C# — язык, весьма похожий на Java. Microsoft также предлагает программу, которая конвертирует Java-код в код на C#.

Говорят, что в настоящее время ведутся работы по созданию CLR для Linux. И если эти разговоры небеспочвенны, то появляется возможность работы .Net-приложения на Linux. И тогда основное преимущество Java, заключающееся в платформенной независимости сходит на нет.

## 6. .Net против J2EE

.Net и J2EE имеют отличия в технических деталях, но в этой статье мы не будем копаться в мелочах. Вместо этого мы рассмотрим два наиболее существенных отличия, а именно технологии реализации многоплатформенности и многоязыковой поддержки.

- **Многоплатформенность** Для разработчика важно то, что и .Net, и J2EE предоставляют средства для создания Web-служб. До сих пор J2EE могла гордиться своей поддержкой многоплатформенности, но, если верить Microsoft, более это не является прерогативой J2EE. Microsoft позиционирует .Net как платформу с двухступенчатой компиляцией, что позволяет создавать среду выполнения для любой платформы, подобно Java. Eric Rutter, старший вице-президент Microsoft запустил информацию о переносе .Net на FreeBSD. FreeBSD — это операционная система, производная от BSD (Berkeley Software Distribution) Unix. Он объявил, что Microsoft в самом деле занимается созданием среды выполнения для FreeBSD. Создание такой среды нарушило бы гегемонию Java в области платформенной независимости, однако не стоит пока полагаться на эту информацию. Создание CLR для наиболее популярных операционных систем может занять много лет. Более того, однажды, Microsoft уже делала подобные заявления в отношении переноса DCOM (Distributed Component

## *Смертельная схватка: .Net против J2EE*

Object Model) на другие платформы. Однако такой перенос так и не был осуществлен.

Таким образом, на сегодняшний день единственной средой разработки, поддерживающей многоплатформенность, является J2EE.

- **Многоязыковая поддержка** Единственной языковой основой J2EE является Java, что сильно отличается от .Net, где поддерживается более двух дюжин языков, включая Fortran, COBOL, C++ и Visual Basic. Можно, конечно, поспорить насчет того, что единственный язык является более элегантным решением, однако надо признать, что .Net предлагает более простое решение для организаций, которые хотят пользоваться знаниями, которые уже имеются у их разработчиков. Ведь для тех разработчиков, которые используют языки, отличные от Java, .Net дает возможность создавать Web-службы на привычном им языке с минимальными затратами на переобучение.

Стратегия Microsoft состоит в том, чтобы рассматривать Java всего лишь как один из языков программирования. Sun в ответ на это заявляет: Java – это не язык программирования, а платформа.

В продолжение темы смотрите послесловие к данной статье — "Microsoft и Sun сравнивают .Net и J2EE", где оба производителя обсуждают сильные и слабые стороны двух платформ.

## **7. Тестируем обе платформы**

В то время, как вокруг .Net и J2EE бушуют споры, мы должны четко понимать, что представляет собой каждая из платформ. Каждая из технологий позволяет создавать Web-службы, доступные потребителям, использующим самые разные технологии. Web-служба на платформе .Net может вызывать J2EE Web-службу и наоборот, поскольку все Web-службы удовлетворяют одинаковым стандартам. Более того, даже если выяснится, что одна из технологий имеет преимущество, то эффективно использовать это преимущество сможет только высоко квалифицированный программист. Инструменты и средства хороши настолько, насколько хорош программист, их использующий.

Если рассуждать с точки зрения развития отрасли программного обеспечения, то некоторые полагают, что результат соревнования технологий .Net и J2EE будет зависеть от работы отделов маркетинга. Эти технологии настолько похожи, что

разница между ними отнюдь не является существенной.

Компания Sun могла бы припомнить высказывание своего пионера Джеймса Гослинга, который на последней Конференции InfoWorld Next-Generation Web Services произнес замечательные дальновидные слова:

Web-службы создают однородное представление для неоднородной среды.

И если это верно, то Web-службы должны прекратить споры о том, какая из платформ является лучшей; разработчикам нет нужды беспокоиться о платформе, разрабатывая сервисы, поскольку Web-службы обеспечивают однородное представление.

В завершение мы предлагаем вам попробовать разработать Web-службы и с помощью .Net, и с помощью J2EE. Тогда вы быстро поймете, какая из платформ лучше подходит для вас. Выбирая поставщика, многие используют такое эмпирическое правило: если в настоящий момент вы используете платформу Microsoft, то выбирайте .Net, в противном случае – J2EE.

## 8. Об авторах

Jonathan Lurie в течении почти десяти лет занимается разработкой коммерческих информационных систем. С 1997 года он использует технологию Java. Он имеет звание бакалавра компьютерных наук, а также целый ряд сертификатов. Во время, свободное от разработки новых программных систем, он занимается преподавательской деятельностью и написанием статей.

R. Jason Belanger более пяти лет занимается разработкой Web-приложений. Он разработал полдюжины полномасштабных коммерческих Web-сайтов. В настоящий момент он руководит компанией NoFeeListing.com.

## 9. Ресурсы

- Определение Web-службы фирмы Sun можно найти в официальном документе, который по ее поручению написал James Kao из компании The Middleware Company: "Developer's Guide to Building XML-Based Web Services with the Java 2 Platform, Enterprise Edition (J2EE)" (TheServerSide.com, June 2001):  
<http://www.theserverside.com/resources/article.jsp?l=WebServices-Dev-Guide>

## Смертельная схватка: .Net против J2EE

- Определение Web-службы фирмы Microsoft можно найти в документе "Defining the Basic Elements of .Net" (Microsoft, 2002):  
<http://www.microsoft.com/net/whatis.asp>
- Лицензии J2EE:  
<http://java.sun.com/j2ee/licensees.html>
- J2EE-совместимые реализации:  
<http://java.sun.com/j2ee/compatibility.html>
- "Shadow Initiatives: .Net and Java," Larry Seltzer (*ZDNet*, January 2002):  
<http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2842294,00.html>
- Недавно Microsoft опубликовала технический обзор, в котором приложение Sun «Pet Store» было переработано с использованием .Net. Хотя это и любопытный документ, необходимо тщательно взвешивать все, что публикует сам поставщик предлагаемой технологии:  
<http://www.gotdotnet.com/team/compare/petshop.aspx>
- The Java Transaction Service:  
<http://java.sun.com/products/jts/index.html>
- "C#: A Language Alternative or Just J--?" Mark Johnson (*JavaWorld*):
  - [Part 1. What the new language for .Net and post-Java Microsoft means to you](#) (November 2000)
  - [Part 2. An in-depth look into the semantic differences and design choices between C# and Java](#) (December 2000)
- Выберите наилучшую стратегию использования Web-служб с помощью статьи "A Birds-Eye View of Web Services," Frank Sommers (*JavaWorld*, January 2002):  
<http://www.javaworld.com/javaworld/jw-01-2002/jw-0125-webservices.html>
- "Sun Adds Web Services to J2EE," Matt Berger, IDG News Service (*JavaWorld*, December 2001):  
<http://www.javaworld.com/javaworld/jw-12-2001/jw-1221-iw-jxml.html?>
- Зайдите в раздел Java 2 Platform, Enterprise Edition тематического каталога JavaWorld:  
[http://www.javaworld.com/channel\\_content/jw-j2ee-index.shtml](http://www.javaworld.com/channel_content/jw-j2ee-index.shtml)
- Зайдите в раздел Java and Web Services тематического каталога JavaWorld:  
[http://www.javaworld.com/channel\\_content/jw-webserv-index.shtml](http://www.javaworld.com/channel_content/jw-webserv-index.shtml)
- Зайдите в раздел Servlets тематического каталога JavaWorld:  
[http://www.javaworld.com/channel\\_content/jw-servlets-index.shtml](http://www.javaworld.com/channel_content/jw-servlets-index.shtml)

- Обсудите Web-службы на форуме Enterprise Java:  
<http://forums.devworld.com/webx?50@@.ee6b80a>
- Подпишитесь на еженедельную бесплатную рассылку журнала JavaWorld Enterprise Java:  
<http://www.javaworld.com/jw-subscribe>
- Огромное количество статей по информационным технологиям вы найдете на [IDG.net](http://IDG.net)

## 10. Microsoft и Sun сравнивают .Net и J2EE

В январе 2002 года мы задали несколько вопросов Microsoft и Sun. Вот, что они нам рассказали:

**Jonathan Lurie:** *В чем сходство .Net и J2EE?*

**John Montgomery, компания Microsoft , руководитель группы, занимающейся платформой .Net:** Сходство спецификаций J2EE и платформы Microsoft .Net в том, что они позволяют упростить и ускорить процесс разработки бизнес-приложений. Microsoft .Net – это платформа, включающая серверы, клиенты и сервисы. Она содержит набор приложений, таких, как Visual Studio .Net, Tablet PC, и .Net My Services. Microsoft .Net была спроектирована таким образом, чтобы удовлетворять текущие и будущие требования заказчиков к созданию, внедрению и управлению приложениями. Фундаментальным принципом Microsoft .Net является изменение процесса разработки и внедрения программного обеспечения – в частности, переход от разработки собственных программ к покупке, настройке и интеграции готового программного обеспечения. Microsoft .Net была разработана для интеграции приложений с помощью Web-служб, используя протоколы и форматы, такие, как SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) и UDDI (Universal Description, Discovery, and Integration).

Отличие J2EE в том, что это всего лишь набор спецификаций. Они определяют лишь небольшую часть полноценной платформы, сфокусированную на разработке приложений на стороне сервера. Эти спецификации, такие как JSP и EJB являются клонами технологий операционной системы Microsoft Windows 2000.

Например, JSP – это прямой клон технологии Microsoft Active Server Pages, а EJB —

## Смертельная схватка: .Net против J2EE

это клон COM+ технологий в Windows. J2EE — это в значительной степени набор спецификаций, спроектированных для облегчения разработки серверных приложений на Unix-системах.

**David F. Harrah, компания Sun Microsystems, менеджер по маркетингу программного обеспечения, основанного на Java и XML:** Главным образом, обе технологии предоставляют платформы для разработки и внедрения программного обеспечения, которые сочетают объектно-ориентированный язык и среду для выполнения приложений. В случае Java, программа на языке Java компилируется в байт-код, который выполняется под управлением Java Runtime Environment (JRE). Код на языках, поддерживаемых .Net, транслируется в Microsoft Intermediate Language, который интерпретируется с помощью среды Common Language Runtime (CLR) в исполняемые инструкции для конкретной платформы.

Новый язык Microsoft C# повторяет Java. J2EE – это реализация Java-технологии, которая дополняет базовые возможности enterprise-компонентами, такими, как EJB и JSP.

Обе технологии созданы для использования разработчиками, создающими приложения, предназначенные для работы в сети.

**Lurie:** *В чем разница между .Net и J2EE?*

**Montgomery:** Прежде всего, J2EE – это набор спецификаций фирмы Sun, выпущенный до пришествия Web-служб – новой технологии, с энтузиазмом встреченной в отрасли программного обеспечения. Microsoft .Net — это набор реализаций, основанный на открытых стандартах, таких как SOAP, WSDL, C#, и CLI (command line interface). Кроме того, J2EE это набор спецификаций, ориентированный на работу серверной части приложений, в то время как Microsoft .Net предлагает полноценную платформу. Наконец, J2EE до сих пор не предлагает стандартного API для работы с XML Web-службами. В отличие от J2EE, с помощью Microsoft .Net, XML Web-службы становятся естественным образом встроенными в платформу. (*Примечание редактора:* Уже после выхода этой статьи Sun выпустила Java Web Services Developer Pack: <http://java.sun.com/webservices/downloads/webservicespack.html>.)

Второе и главное крупное отличие состоит в том, что Microsoft .Net спроектирована для поддержки нескольких языков программирования – в настоящее время, Microsoft

.Net поддерживает более 20 языков, позволяя создавать .Net приложения на любых языках без затрат на переобучение персонала. Что же касается J2EE, то эта платформа поддерживает единственный язык программирования — Java.

**Harrah:** Прежде всего, Java — это результат взаимодействия более чем 400 компаний и организаций, а .Net – это продукт одной компании. Технология Java поддерживается и совершенствуется с помощью так называемого процесса Java Community Process (JCP), представляющего из себя взаимодействие группы из более чем 400 компаний, организаций и частных лиц в целях создания платформы для сервисов и приложений, которая может работать на системах любого типа. Java Community Process предполагает создание групп экспертов, состоящих из заинтересованных членов, которые сотрудничают в целях определения новых спецификаций и усовершенствования существующих. Система принятия решений с помощью голосования гарантирует, что Java остается единой и общей платформой для всех без каких-либо предпочтений для какой-то одной компании.

Java приложения могут выполняться на любых операционных системах: системах уровня предприятия, таких, как Unix, Linux, OS/390, Windows 2000, или HP-UX; операционных системах для десктопов, таких как Mac OS, Windows или Linux; а также операционных системах для мобильных устройств, таких как Palm OS или Symbian's EPOC. .Net была полностью разработана Microsoft и может работать только на операционных системах Microsoft.

Технология Java является открытой и построена на внутриотраслевых стандартах для программного обеспечения. Любой желающий может загрузить и изучать код Java платформы. Microsoft приоткрыла лишь небольшие части технологии .Net, такие как язык C#, но повесила железный занавес на ключевые области своей платформы и не публикует их в открытую. Microsoft лишь избирательно делает небольшие части своего исходного кода доступными для отдельных партнеров.

.Net – это новая платформа Microsoft, и на сегодняшний день доступны лишь ее бета-версии. Технология Java активно используется в течение почти 6 лет, что означает, что в различные редакции компонент Java заложен значительный опыт использования платформы. В настоящий момент доступна уже третья версия J2EE в то время как Java Community Process занят разработкой четвертой. (Примечание редактора: Уже после выхода данной статьи Microsoft выпустила Microsoft Visual

## *Смертельная схватка: .Net против J2EE*

Studio .Net: <http://msdn.microsoft.com/vstudio/>.)

**Lurie:** *Какие преимущества имеет J2EE перед .Net?*

**Montgomery:** Главное преимущество J2EE в том, что можно приобрести различные реализации базовых спецификаций у различных поставщиков.

**Harrah:** Технология Java изначально создавалась в расчете на использование в сетевых приложениях. Она упрощает гетерогенную природу сети и поддерживается на любых операционных системах и микропроцессорных архитектурах, которые только есть в современных сетях. С самого начала в Java была заложена строгая и надежная модель безопасности, поэтому Java уязвима для хакеров и вирусов значительно меньше, чем продукты Microsoft.

Java поддерживает не только Web-службы, но и другие виды служб, такие как беспроводные службы данных (wireless data services) и сервисы, активизируемые по требованию (services on demand). Java поддерживает такие связанные с Web-службами технологии, как XML, SOAP и UDDI и по опросу Evans Data Corp. Developer является платформой номер один для построения Web-служб. Я не слышал, чтобы Microsoft внятно объяснила, каким образом она поддерживает wireless data services в .Net. И это в то время, когда у пользователей имеется по крайней мере 15 миллионов поддерживающих Java телефонов и использующих wireless data services; особенно это актуально для Японии.

**Lurie:** *Какие преимущества имеет .Net перед J2EE?*

**Montgomery:** Главное преимущество Microsoft .Net в том, что это полноценная платформа, а J2EE ориентирована только на серверное программирование. Более того, J2EE — это лишь набор спецификаций и необходимо приобретать дорогостоящие (обычно порядка \$15,000 для одной машины) реализации J2EE. В отличие от J2EE, Microsoft .Net – это набор продуктов и служб. В дополнение к этому, Microsoft .Net имеет встроенные в платформу XML Web-службы, а не просто использует их, как дополнительно подключаемый механизм. Это позволяет существенно увеличить производительность как самих приложений, так и труда разработчиков. Microsoft .Net разрабатывался с поддержкой интеграции посредством XML Web-служб с использованием протоколов и форматов таких, как SOAP, WSDL и UDDI.

## Смертельная схватка: .Net против J2EE

Как я уже упоминал выше, Microsoft .Net поддерживает различные языки программирования; J2EE поддерживает единственный язык: Java.

**Harrah:** Я не вижу НИКАКИХ преимуществ.

Reprinted with permission from the March 2002 edition of JavaWorld magazine. Copyright © ITworld.com, Inc., an IDG Communications company.

View the original article at: <http://www.javaworld.com/javaworld/jw-03-2002/jw-0308-j2eenet.html>

[Перевод на русский © Константин Разумовский. 2002](#)